**Лабораторна робота №14. Застосування технологій конструювання програмного забезпечення. Прототипування програмних систем та розробка користувальницького інтерфейсу**

**Мета роботи**. Набуття навичок проектування графічного інтерфейсу користувача з використовуванням прототипування

**Завдання на лабораторну роботу**

1. Ознайомитись з теоретичними відомостями, необхідними для виконання роботи.

2. Для власного курсового проекту потрібно створити прототип графічного інтерфейсу користувача. Визначте:

2.1. складові інтерфейсу користувача;

2.2. перелік графічних елементів необхідних для реалізації графічного інтерфейсу;

2.3. визначте та обґрунтуйте підходи до проектування ПЗ щодо інтерфейсу користувача для Вашої курсової роботи;

2.4. визначте стиль інтерфейсу користувача;

2.5. визначте та обґрунтуйте обрану модель інтерфейсу користувача;

2.6. надайте схематично вигляд первісного прототипу інтерфейсу користувача;

2.7. визначте етапи розробки ПЗ, на яких планується залучення користувача до розробки інтерфейсу та надання йому чергової версії прототипу. Вкажіть які елементи та функції Ви плануєте вдосконалювати/впроваджувати на цьому етапі.

3. Оформити звіт, де надати у вигляді таблиць результати за пп.2.1-2.2, додавши відповідні пояснення щодо обраного рішення.

**Контрольні запитання**

1. Поясніть, чому для розробки великих систем рекомендується експериментальне прототипування.
2. У яких обставинах ви рекомендували б прототипування як засіб обґрунтування системних вимог?
3. Опишіть труднощі, які можуть виникнути при прототипуванні вбудованих комп'ютерних систем реального часу.
4. Як найбільше ефективно визначити придатні для повторного використання компонента?
5. Які переваги і недоліки використання механізму OLE для швидкої розробки додатків?

Результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді текстових файлів з іменем у форматі **KPZ<Номер групи><Номер лекції / лабораторної>[-<Номер завдання>][літера позначення типу роботи L – лекція, R - лабораторна]<Прізвище англійською>**. Наприклад, **KPZ4112R**buts.doc. Відповіді повинні бути не довгими і змістовними. Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності відповідей-"близнюків" відповідь буде зараховуватися першому за часом надсилання.

**Строк відсилки ЛР IПЗ41 - 05.04.2021, IПЗ42 - 01.04.2021.**

**До ЛР№13**

При поданні інформації щодо складових методології (рис.4 ЛР№13) доцільно використовувати групування в таблицях (складання таблиць входило у завдання).

|  |  |  |
| --- | --- | --- |
| Групи | Ролі | Уміння |
| 1 | 2 | 3 |
|  |  |  |
|  |  |  |

В графі 1 визначаєте групи із визначенням їх ролей (наприклад, дизайнери, розробники документації, тестери та ін.), в графі 2 перелік посад по цих групах (наприклад, менеджер проекту, тестувальник, розробник інтерфейсу та ін.), в графі 3 навички, які повинні мати претенденти на посаду (у відповідному рядку, наприклад, розробник інтерфейсу – знання HTML, CSS тощо).

Аналогічні таблиці доцільно скласти для придання інформації щодо Видів діяльності – Технік – Інструментів, а також Якості – Продуктів – Стандартів.

**ТЕОРЕТИЧНІ ВІДОМОСТІ**

Графічні інтерфейси користувача в цей час є стандартом для інтерактивних систем. Зусилля, що вкладаються у визначення, проектування і реалізацію такого інтерфейсу, складають значну частину вартості розробки додатка. Розробники не повинні нав'язувати користувачам свою точку зору на проектований інтерфейс. Користувачі повинні брати активну участь у процесі проектування інтерфейсу. Такий погляд на розробку інтерфейсу привів до підходу, названого проектуванням, орієнтованим на користувача (user-centred design), який заснований на прототипуванні інтерфейсу і участі користувача в процесі його проектування.

У цьому аспекті прототипування ‒ необхідна частина процесу проектування користувацького інтерфейсу. Через динамічну природу користувацьких інтерфейсів текстових описів і діаграм недостатньо для формування вимог до інтерфейсу. Тому еволюційне прототипування за участю кінцевого користувача ‒ єдиний прийнятний спосіб розробки графічного інтерфейсу для програмних систем.

Генератори інтерфейсів - це графічні системи проектування екранних форм, де інтерфейси компонуються з елементів типу меню, полів, піктограм і кнопок, які, у свою чергу, можна просто вибрати з меню і помістити в екранну форму. Системи цього типу ‒ необхідна частина систем програмування баз даних. Генератори інтерфейсів створюють добре структуровану програму, згенеровану по специфікації інтерфейсу.

Мільйони людей сьогодні мають доступ до Web-броузерів. Вони підтримують мову розмітки гіпертексту HTML, який дозволяє створювати користувацькі інтерфейси. Кнопки, поля, форми і таблиці можуть бути включені до Web-сторінки так само, як і засоби мультимедіа. Сценарії обробки подій і даних, пов'язані з об'єктами інтерфейсу, можуть виконуватися або на машині Web-клієнта, або на Web-сервері.

Через широкі можливості Web-броузерів і потужності мови HTML у цей час усе більше користувацьких інтерфейсів будуються як Web-орієнтовані. Такі інтерфейси ‒ приналежність не тільки нових систем; вони заміняють інтерфейси, побудовані на текстових формах, у широкому колі наслідуваних систем.

Для Web-орієнтованих інтерфейсів прототипи можна створювати за допомогою стандартних редакторів Web-сторінок, які, по суті, будують користувацькі інтерфейси. Об'єкти на Web-сторінці визначаються, як і пов'язані з ними операції, за допомогою вбудованих засобів мови HTML (наприклад, зв'язування з іншою сторінкою) або за допомогою мови Java або сценаріїв.

Інтерфейс користувача (ІК) - це комплекс апаратних і програмних засобів, що забезпечує взаємодію користувача з комп'ютером. ІК комп’ютерного додатку включає:

1) засоби відображення інформації, відображувану інформацію, формати і коди;

2) командні режими, мову «користувач–інтерфейс»;

3) пристрої та технології введення-виведення;

4) діалоги, взаємодію та транзакції між користувачем та комп’ютером, зворотній зв’язок з користувачем;

5) підтримку прийняття рішень в конкретній предметній області;

6) порядок використання програми і документації на неї.

ІК часто розуміють лише як зовнішній вигляд програми. Однак насправді користувач сприймає через нього всю програму в цілому, тобто таке розуміння є надто вузьким.

В дійсності, ІК об’єднує в собі всі елементи і компоненти програми, які здатні впливати на взаємодію користувача з програмним забезпеченням (рис.1.).

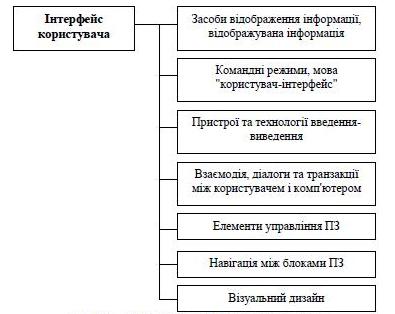
****

Рисунок 1. Складові інтерфейсу користувача

Грамотно спроектований інтерфейс користувача украй важливий для успішної роботи системи. Складний в застосуванні інтерфейс, як мінімум, приводить до помилок користувача. Іноді вони просто відмовляються працювати з програмною системою, не дивлячись на її функціональні можливості.

Таблиця 1. Елементи графічних інтерфейсів користувача

|  |  |
| --- | --- |
| Елементи | Опис |
| Вікна | Дозволяють відображати на екрані інформацію різного роду |
| Піктограми | Представляють різні типи даних. У одних системах піктограми представляють файли, в інших – процеси |
| Меню | Введення команд замінюється вибором команд з меню |
| Вказівники | Миша використовується як пристрій вказівки для вибору команд з меню і для виділення окремих елементів у вікні |
| Графічні елементи | Можуть використовуватися спільно з текстовими |

На рис. 2 зображений ітераційний процес проектування призначеного для користувача інтерфейсу. На початку процесу прототипування створюються паперові макети інтерфейсу, потім розробляються екранні форми, що моделюють взаємодію з користувачем.



Рисунок 2. Процес проектування інтерфейсу користувача

Важливим етапом процесу проектування інтерфейсу користувача є аналіз діяльності користувачів, яку повинна забезпечити обчислювальна система. Не вивчивши того, що, з погляду користувача, повинна робити система, неможливо сформувати реалістичний погляд на проектування ефективного інтерфейсу. Для аналізу потрібно застосовувати різні методики, а саме: аналіз завдань, етнографічний підхід, опити користувачів і спостереження за їх роботою.

**Труднощі проектування ІК**

Сучасне середовище розробки ПЗ містить в собі велику кількість проблем: необхідність зниження витрат, скорочення термінів, розробки більш передбачуваних планів, надання більш якісних розв’язків, забезпечення нескладного в засвоєнні та використанні ПЗ, постійне оволодіння новими технологіями та засобами, досягнення кращих результатів в порівнянні з конкурентами, а також інші аспекти.

Задоволеність користувача програмним продуктом або його практичністю може бути в значній мірі віднесена на рахунок ІК. Взагалі, задоволеність користувача – це функція невеликої кількості факторів:

**задоволеність користувача =** функція від МОЖЛИВОСТЕЙ ІНТЕРФЕЙСУ КОРИСТУВАЧА, ЧАСУ ВІДГУКУ, НАДІЙНОСТІ, пристосованості до інсталяції, інформаційної підтримки, пристосованості до супроводження та інших факторів.

Фактори, виділені в рівнянні великими літерами, відіграють найбільш важливу роль. Головне, на що звертають увагу користувачі – швидке, легке та надійне виконання роботи за допомогою засобів, які автоматизують, доповнюють та полегшують виконання задачі. Простота інсталяції/деінсталяції має важливе значення, оскільки вона задає тон на початку та в кінці досвіду спілкування користувача з програмою. Легкість оновлення ПЗ має таке ж важливе значення, як і інші інтерактивні властивості продукту. Інформаційна підтримка (навчання, настанови, довідки та підтримка експлуатації) важлива під час початкового або наступного освоєння продукту, але не має безпосереднього відношення до розв’язуваної задачі.

Існує також багато інших факторів (узгодженість, інтеграція, вартість), які впливають на рівень задоволеності користувача в залежності від їх значення, середовища, розв’язуваної задачі та ситуації.

В ході планування розробки продукту, аналізу вимог, проектування, реалізації, тестування і розгортання потрібно враховувати всі фактори, які входять в рівняння задоволеності користувача та їх відносну важливість.

Користувач або представник компанії не завжди може чітко визначити свої потреби та вимоги на етапі планування та аналізу вимог, однак він може бути дуже категоричним відносно того, що бажано й небажано в програмному продукті, коли він вже стане доступним для апробації та використання.

Ділові й функціональні потреби – найбільш загальний вид вимог. Однак вимоги до характеристик ІК, практичності, інтеграції та узгодженості дуже часто пропускаються. В багатьох випадках характеристик ІК визначаються неявно (може навіть несвідомо) у вигляді деяких очікувань. Враховуючи, що приблизно 50% програмного коду приходиться на ІК, для розуміння вимог до видимих властивостей продукту потрібно докласти додаткових зусиль. Вимоги до практичності, інтеграції та узгодженості повинні бути визначені явно і піддаватись вимірюванню. Лише тоді можна перевірити й виміряти, наскільки продукт відповідає планам. Після встановлення вимог, їх необхідно контролювати і управляти ними.

Щодо термінів розробки, то це питання кваліфікації та досвіду стосовно до сучасної технології ІК. З множиною стилів та функцій ІК пов’язаний великий обсяг інформації, який включає масу деталей, особливості очікуваної поведінки, а також надлишкові відомості.

**Проектування ІК з орієнтацією на користувача.**

Програмний ІК привертає все більшу увагу і набуває все більшого значення як складова конкурентної переваги. По мірі того, як перелік функцій програмних засобів стає все довшим та складнішим, користувачі, які відповідають за придбання продукту, дивляться на ІК як на вирішення проблеми складності. Якщо ІК продукту привертає увагу користувача, якщо він легкий у вивченні, простий у використанні, а також має прийнятну ціну та можливості, продукт має конкурентну перевагу. Конкурентну перевагу можна одержати в тому випадку, якщо заявки про нижчі витрати на навчання та виграш в продуктивності відповідають дійсності.

Існує декілька умов, які дозволяють говорити про те, що проект ведеться в орієнтованому на користувача стилі:

1) розуміння користувачів та їх задач, залучення користувачів в усі аспекти життєвого циклу продукту;

2) постановка цілей, які можна виміряти; встановлення критеріїв успіху з точки зору користувачів та підприємств;

3) проект повинен передбачати нову компетентність користувача, яка по відношенню до продукту включає пакетування, маркетинг, навчання, віддруковану інформацію, налагодження параметрів, інсталяцію, екрани, графіку, довідки, іншу експлуатаційну підтримку, оновлення та деінсталяцію;

4) оцінювання та тестування за участю реальних користувачів для визначення, чи досягнуті цілі та які проблеми існують;

5) ітеративний підхід – якщо цілі не досягнуті або існують проблеми, потрібно внести виправлення та провести повторну перевірку.

Система та її ПЗ спілкуються з користувачем мовою представлення візуальної та слухової інформації, а також на рівні невербальної регламентації взаємодії, яка виражається у вигляді часу відзиву, надійності, поведінки ті інших людських факторів. Користувач спілкується з системою та її ПЗ мовою дій за допомогою клавіатури, миші, мовної та тактильної інформації. Іноді система або її ПЗ накладають на користувача обмеження, пов’язані з введенням інформації в межах певного проміжку часу (час відзиву), безпомилковим спілкуванням (надійність) та коректним підходом до використання (поведінка).

В традиційних процесах проектування використовуються різні підходи. Основні змінні, пов’язані з процесом розробки, характеризують ступінь, в якій цей процес може бути віднесений до одного з типів. Це, наприклад, такі типи процесу проектування, як проектування “ззовні-всередину” (outside-in) або “зсередини-назовні” (inside-out); однократне (без ітерацій) або багатократне (ітеративне) проектування; проектування за типом “великого вибуху” (big bang, “все або нічого”) або еволюційне (табл.2).

Таблиця 2. Підходи до проектування ПЗ

| **№** | **Підхід до проектування** | **Опис підходу** |
| --- | --- | --- |
| 1 | Проектування "ззовні-всередину" (outside-in) | Спрямований на інтерфейс та доступні користувачу властивості програмного продукту |
| 2 | Проектування "зсередини-назовні" (inside-out) | Починається з розгляду внутрішніх властивостей системи |
| 3 | Однократне проектування (без ітерацій) | Проектування без встановлення планованого обсягу робіт з конструювання продукту та ІК |
| 4 | Багатократне (ітераційне) проектування | Концентрується на ітераційній побудові ІК та його основних факторах практичності |
| 5 | Проектування з використанням теорії "великого вибуху" (big bang theory) | Спроба розробити "все або нічого", тобто ПЗ проектується і реалізується паралельно |
| 6 | Еволюційне проектування | Зосереджене на побудові продукту з покроковим нарощуванням і уточненням можливостей продукту |

Найкращий підхід до розробки – еволюційний ітеративний підхід “ззовні всередину”.

*Орієнтоване на користувача проектування продукту (UCD – User-Centered Design)* – міждисциплінарний та ітеративний процес розробки ПЗ, спрямований на досягнення користувацьких цілей у відношенні продукту, на практичність та інші вимірювані властивості продукту на протязі його життєвого циклу.

Навіть для відносно невеликих проектів задача проектування програмного ІК є доволі непростою, не говорячи вже про розробку складного пакету програмних додатків з застосуванням нових технологій в умовах потужних конкурентних факторів жорстких проектних обмежень. Сьогодні для створення стилів ІК та додатків неодмінно потрібен широкий набір різноманітних спеціальних навичок в таких областях, як користувацький інтерфейс, розробка ПЗ, тестування і якість функціювання ПЗ, графіка і візуалізація, навчання, довідкові системи, підтримка експлуатації, людські фактори, бізнес-планування, управління потоками робіт, бізнес-реінжиніринг, управління проектами та управління змінами.

Успішність побудови ІК та програмного продукту полягає в професіоналізмі розробників, які повинні володіти наступними здібностями:

1) вміти працювати в команді;

2) працювати з різними людьми як в організаціях, які займаються розробкою, так і в комерційних організаціях;

3) розуміти існуючі запити користувачів, а також їх майбутні потреби й цілі;

4) вміти уточнювати й розширяти до необхідного ступеня деталізації неясно сформульовані вимоги;

5) використовувати засоби й методи, які необов’язково призначені для розробки ПЗ;

6) швидко й ефективно оцінювати й відновлювати проект та реалізацію.

Проблеми організаційного, групового та індивідуального характеру, які виникають при проектуванні продукту, відрізняються набагато більшою складністю, ніж технічні проблеми. 80% проблем пов’язані з людьми і лише 20% носять технічний характер.

Головна мета проектного колективу полягає в ефективній поставці продукту, який задовольняє вимогам до практичності та узгодженості. Крім того, колектив вимагає чіткого управління проектом, підтримки керівництва, інструментальних засобів та вірного настрою, щоб впоратись з нелегкою роботою.

Значення різних професійних навичок та участь у проекті відповідної кількості спеціалістів, які володіють кожним типом навичок, потребує ретельного планування для досягнення успішності проекту. Правильне визначення ролі розробника в колективі розробників дозволяє усунути будь-які невизначеності. Одна людина може виконувати одну або декілька ролей. Кожен спеціаліст повинен володіти достатніми професійними навичками і бути готовим, згодним та здатним виконувати декілька ролей.

Для успішного створення продукту найважливіше значення має висококваліфікований персонал, який володіє навичками в таких областях, як: технологія розробки ІК, стандарти ІК, інструментальні засоби реалізації та використання ІК в додатках; проектування та реалізація ПЗ ІК; проектування та реалізація візуальних та графічних конструкцій; проектування та реалізація засобів навчання, керівництв, систем допомоги та електронної експлуатаційної підтримки; психологія та ергономіка, людська поведінка, сприйняття, навчання та пізнання; складання реальних бізнес-процесів.

**Етапи проектування ІК, орієнтованого на користувача:**

1) план – план створення продукту сконцентрований на побудові ІК та забезпеченні практичності; план враховує календарні терміни для кожного з етапів процесу, визначає основні фактори ризику, об’єднує всі можливі методи, встановлює цілі та критерії по відношенню до ІК;

2) вимоги – на етапі встановлення вимог виконуються наступні задачі: опис користувачів, постановка задач користувачів, оцінка поточного рівня практичності, аналіз можливостей ІК, аналіз тенденцій;

3) концептуальне проектування – концептуальний проект представляє собою сукупність високорівневих описів, абстракцій та оглядової інформації, яка дає розробникам та кінцевим користувачам загальне уявлення про програмний продукт, його структуру та ІК;

4) проектування – проект ІК представляє сукупність характеристик програми, які сприймаються користувачем (вхідні сигнали, взаємодія користувача, відзив системи на вхідні сигнали та взаємодію користувача);

5) прототипування – створення прототипів та моделювання – ефективні засоби ранньої оцінки проекту; прототип – це матеріалізація побудованого проекту з використанням його передбачуваної платформи реалізації, включаючи обладнання, ОС, мови і засоби реалізації;

6) специфікація – матеріалізація проекту програмного продукту в документальній формі, яка описує і показує дії користувачів, а також вигляд та поведінку ПЗ в специфічних ситуаціях;

7) конструювання – написання коду та автономне тестування;

8) оцінка – всі методи оцінки пов’язані з залученням потенційних користувачів програмного продукту;

9) ітеративний підхід – загальні критерії досягнення цілей створення ІК повинні бути чітко визначені, зрозумілі й прийняті керівництвом та розробниками; досягнення поставлених цілей може вимагати багатократних ітерацій;

10) розгортання – після того, як продукт задовольняє вимогам та потребам користувачів, він розгортається для використання за призначенням; з цього моменту починається ряд наступних дій – оцінка продукту за участю користувачів, які не залучались до розробки, пілотне тестування, виконання користувацьких задач, які не оцінювались або не були передбачені під час проектування та розробки.

*Засоби проектування:*

- програмне та апаратне забезпечення;

- обладнання;

- матеріали;

- шаблони проектування / прототипування / оцінки;

- засоби, необхідні для проекту, інструментальні засоби розробки.

**Інтерфейс меню**

Меню являють собою перелік можливих дій, відображуваних на екрані або в вікні для користувачів, з метою вибору потрібних варіантів. Меню дозволяють користувачам пересуватись всередині системи, надаючи маршрути і можливість вибору потрібних пунктів або варіантів виконання операцій. Меню є дуже важливою частиною ІК, як графічного, так і об'єктно-орієнтованого. Стилі меню розвиваються.

Меню бувають різними за формою, розмірами і стилем. При *повноекранному меню* (доступні функції або задачі представлені у вигляді списку на екрані. Основною визначальною характеристикою більшості повноекранних інтерфейсів меню є їх ієрархічна будова, тобто при виконанні задачі користувачам необхідно пересуватись деревовидною структурою меню.

Більшість сучасних графічних програм використовують *панель меню* (рис.3), розташовану в верхній частині екрану або вікна. Такий тип меню забезпечує швидкий доступ до меню, а користувачі працюють всередині екранів або вікон програм. Панель меню представляє собою динамічний список основного набору операцій або пунктів, які ведуть користувача до інших пунктів, представлених в окремому спадному меню. Панелі меню є невід'ємною частиною всіх основних ГІК і розроблених для них програм.



Рисунок 3. Панель меню та підменю графічного ІК

Основною характеристикою меню є їх здатність динамічно змінюватись. Це забезпечує користувачів підходящими варіантами і маршрутами для виконання поточних задачі або вибору потрібних об'єктів. Пункти меню можуть виділятись сірим кольором, якщо деякі операції на даний момент недоступні. Іншим типом меню, розробленим для середовища ІК, є *спливаюче або контекстне меню*), оскільки його вміст залежить від контексту задач, які стоять перед користувачем. Вони можуть використовуватись для будь-якого типу додатків, як графічного, так і повноекранного інтерфейсу. Спливаючі меню містять лише ті варіанти, які відповідають поточному або обраному пункту. Розробляються вони для будь-якого елементу інтерфейсу. Такі меню містять невеликий набір часто використовуваних дій, які можна знайти і на панелі меню системи чи програми.

*Переваги меню:*

- користувачам не потрібно пам'ятати набір команди;

- відмова від використання клавіатури зменшує кількість помилок;

- навігація по ієрархії проста для нових або випадкових користувачів;

- скорочує час навчання користувача;

- легко відстежувати відповіді і коригувати помилки;

- необхідний мінімальний обсяг друку;

- може бути використане сумісно з іншими типами інтерфейсів;

- елементи меню та їх розташування можуть налагоджуватись користувачем;

- гнучка система вибору команд (мнемоніки, миша, "гарячі" сполучення клавіш);

- підтримує тип пам'яті, орієнтований на взнавання.

*Недоліки меню:*

- невідповідності або неефективність для деяких користувачів або задач;

- потрібна техніка навігації і вибору;

- необов'язково робить інтерфейс зручним;

- займає багато місця на екрані;

- змушує користувача пересуватись через багато рівнів підменю;

- вимагає часто оновлювати екран;

- користувач повинен розуміти принцип групування та ієрархії меню;

- вимагає деяких знань про систему;

- користувач може заблукати в ієрархії меню;

- терміни і найменування в меню незрозумілі користувачу;

- структура меню слідує за синтаксисом "об'єкт-дія";

- синтаксис меню повинен бути послідовним;

- використання режимів змушує користувачів стежити за системою

**Стилі інтерфейсу користувача**

*Графічний інтерфейс (GUI – Graphical User Interface)* – стиль взаємодії “користувач-комп’ютер”, в якому застосовуються 4 елемента: вікна, піктограми, меню та покажчики. Іноді ГІК називають WIMP-інтерфейсом (Windows, Icons, Menus, Pointers).

*Web-інтерфейс користувача (WUI – Web User Interface)* – дуже схожий на меню ієрархічної структури. Необхідна навігація виконується в рамках одного або декількох додатків з використанням текстових або візуальних гіперпосилань. До найбільш поширених компонентів WUI-інтерфейсу відносяться баннери (заголовки), навігаційні панелі та візуальні або текстові гіперпосилання, впорядковані різними способами.

*Інтерфейс користувача кишенькових пристроїв (HUI)* – інтерфейс користувача, для введення даних в якому застосовують “жестикуляційний” стиль з пером та сенсорним маленьким екраном. Загальний стиль ІК для НUI-інтерфейсу можна назвати SIMP-стилем (Screen, Icons, Menus, Pointers).

*Прикладний рівень ІК програмного забезпечення* - спосіб використання GUI-стиля з компонентами ІК прикладного рівня

*Об’єктно-орієнтовані ІК* – має об’єктно-орієнтований стиль та/або об’єктно-орієнтовану внутрішню структуру.

**Моделі інтерфейсу користувача**

*Ментальна, або концептуальна модель* - лише внутрішнє відображення того, як користувач розуміє і взаємодіє з системою. (див. рис.4)

*Модель користувача - с*прийняття роботи програмного забезпечення користувачем. Це погляд користувача на події, які відбуваються в ПЗ.

*Модель програміста* є функціональною специфікацією програмного продукту.

*Модель проектувальника* – це дещо середнє між моделлю користувача та моделлю програміста.



Рисунок 4. Компоненти концептуальної моделі

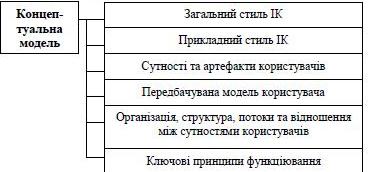


Рисунок 5. Компоненти моделі користувача

***Ключові питання розробки ІК***

До ключових питань розробки можна віднести:

* керуючі елементи;
* рядок меню та панель інструментів;
* метод drag-and-Drop;
* компоновку і розробку вікна.

***Рекомендації при створенні інтерфейсу користувача***:

- Процес проектування інтерфейсу повинен орієнтуватися на користувача. Інтерфейс повинен взаємодіяти з користувачем на його «мові», бути логічним і послідовним. У інтерфейсі повинні бути довідкові засоби, що допомагають користувачам при роботі з системою, і засоби відновлення після помилок.

- Існує декілька стилів взаємодії з програмними системами: безпосереднє маніпулювання, системне меню, заповнення форми, командні мови і природна мова.

- Для відображення тенденцій числових даних і їх приблизних значень потрібно використовувати графічні уявлення. Числове уявлення повинне застосовуватися тільки тоді, коли потрібно відобразити точні значення даних.

- Кольори в інтерфейсі користувача повинні використовуватися обережно і послідовно. Розробники повинні завжди пам'ятати, що багато хто не розрізняє кольори.

- Повідомлення про помилки не повинні містити звинувачень в адресу користувача. Вони повинні пропонувати варіанти виправлення помилки і забезпечувати зв'язок з довідковою системою.

- У документації користувача повинне бути керівництво для початкуючих і досвідчених користувачів. Для системного адміністратора повинні бути окремі документи.

- У системній специфікації бажано мати кількісні значення для показників зручності використання інтерфейсу, а процес його оцінювання повинен перевіряти систему на відповідність цим вимогам.

Основні рекомендації по розробленню інтерфейсу і компонуванню вікна:

* асиметрія = активний стан, симетрія = стан спокою;
* ніколи не треба гнатись за помірністю;
* потрібно уважно стежити за послідовністю та узгодженістю;
* потрібно пам'ятати, що узгоджені інтерфейси означають кращу впроваджуваність системи;
* потрібно чітко вирівнювати керуючі елементи;
* якщо треба розташовувати деякі елементи за межами вирівнювання, то потрібно робити це логічно;
* потрібно створювати екранні об'єкти за зразком об'єктів з реального фізичного світу;
* не потрібно забувати, що абсолютна симетричність ускладнює читання інформації з екрану;
* елементи, які не відповідають шаблону, виділяються з загальної маси;
* елементи однакового розміру та кольору сприймаються як приналежні до однієї групи.